# Titanic

May 2, 2024

## 1 TITANIC E I SOPPRAVVISSUTI

### 1.1 CREATO DA GRANIERI JOELE, TASSONE LEONARDO, RAPHAEL RODRIGO E RADISHA WARNAKULASURIYA

#### 1.1.1 IMPORTAZIONE DEI DATI DA OSSERVARE

[1]:
```
import pandas as pd
titanic = pd.read_csv("C:\\Users\\joele\\OneDrive\\Desktop\\dati␣
 ↪robotica\\gender_submission.csv")
titanic
```

[1]:
```
     PassengerId  Survived
0            892         0
1            893         1
2            894         0
3            895         0
4            896         1
..           …       …
413         1305         0
414         1306         1
415         1307         0
416         1308         0
417         1309         0

[418 rows x 2 columns]
```

[2]:
```
import pandas as pd
titanic = pd.read_csv("C:\\Users\\joele\\OneDrive\\Desktop\\dati robotica\\test.
 ↪csv")
titanic
```

[2]:
```
     PassengerId  Pclass                                          Name  \
0            892       3                              Kelly, Mr. James
1            893       3              Wilkes, Mrs. James (Ellen Needs)
2            894       2                     Myles, Mr. Thomas Francis
3            895       3                              Wirz, Mr. Albert
4            896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)
```

```
..           …     …                                        …
413         1305       3                          Spector, Mr. Woolf
414         1306       1                Oliva y Ocana, Dona. Fermina
415         1307       3                Saether, Mr. Simon Sivertsen
416         1308       3                         Ware, Mr. Frederick
417         1309       3                     Peter, Master. Michael J

        Sex   Age  SibSp  Parch              Ticket      Fare Cabin Embarked
0      male  34.5      0      0              330911    7.8292   NaN        Q
1    female  47.0      1      0              363272    7.0000   NaN        S
2      male  62.0      0      0              240276    9.6875   NaN        Q
3      male  27.0      0      0              315154    8.6625   NaN        S
4    female  22.0      1      1             3101298   12.2875   NaN        S
..      …   …    …    …                 …        …   …      …
413    male   NaN      0      0           A.5. 3236    8.0500   NaN        S
414  female  39.0      0      0            PC 17758  108.9000  C105        C
415    male  38.5      0      0  SOTON/O.Q. 3101262    7.2500   NaN        S
416    male   NaN      0      0              359309    8.0500   NaN        S
417    male   NaN      1      1                2668   22.3583   NaN        C

[418 rows x 11 columns]
```

[3]:
```
import pandas as pd
titanic = pd.read_csv("C:\\Users\\joele\\OneDrive\\Desktop\\dati␣
 ↪robotica\\train.csv")
titanic
```

[3]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …       …     …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                 Name     Sex   Age  SibSp  \
0                             Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                                Heikkinen, Miss. Laina  female  26.0      0
3          Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                              Allen, Mr. William Henry    male  35.0      0
..                                                  …     …   …    …
```

```
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …       …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

### 1.1.2 CONTROLLO DELLE PRIME RIGHE DEI PASSEGGERI

[4]:
```
titanic.head()
```

[4]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

### 1.1.3 CONTROLLO DELL'ETA' DEI PASSEGGERI

[5]:
```
mean_age = titanic['Age'].mean()
titanic['Age'].fillna(mean_age, inplace=True)
titanic['Age']
```

[5]:
```
0      22.000000
1      38.000000
2      26.000000
3      35.000000
4      35.000000
         …
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, Length: 891, dtype: float64
```

qui controlliamo l'età dei passengeri e sostituiamo le età mancanti con .fillna calcolando la media

### 1.1.4 CONTROLLO DEI COSTI DEI BIGLIETTI

[6]:
```
titanic['Fare'].fillna(method='ffill', inplace=True)
titanic['Fare']
```

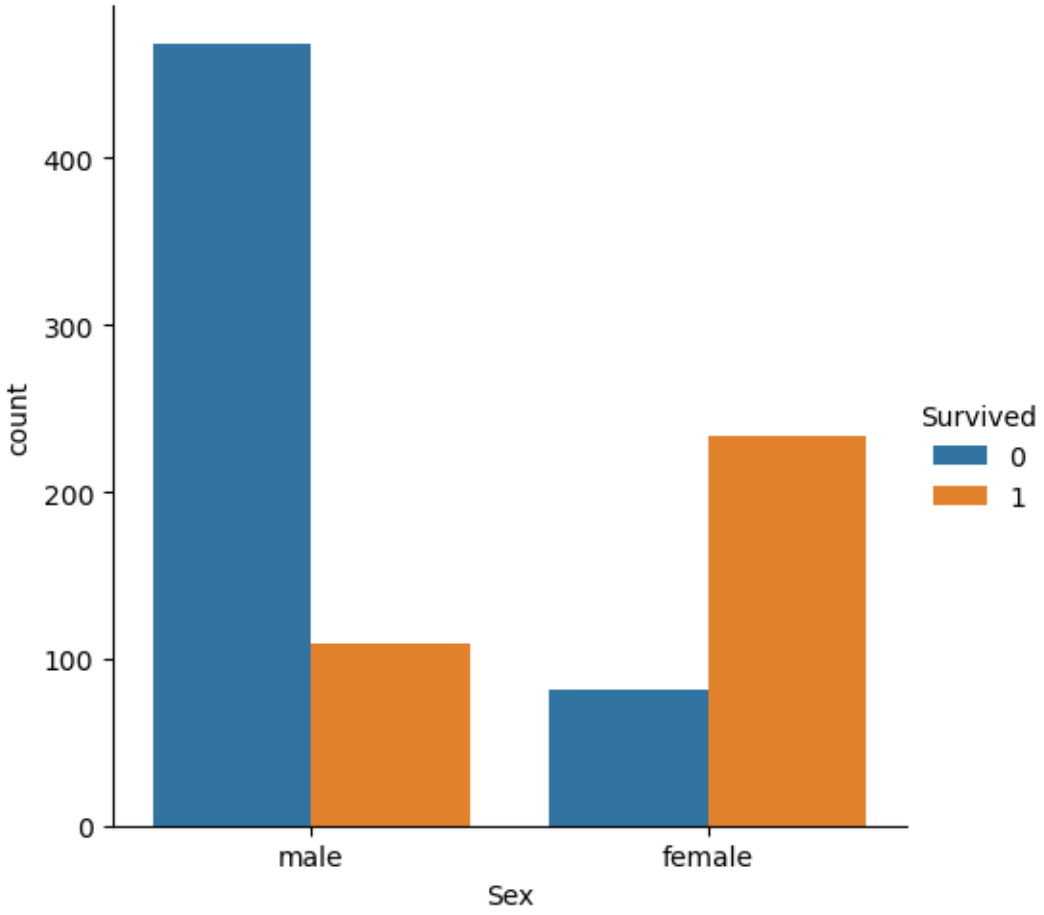
[6]:
```
0       7.2500
1      71.2833
2       7.9250
3      53.1000
4       8.0500
         …
886    13.0000
887    30.0000
888    23.4500
889    30.0000
890     7.7500
Name: Fare, Length: 891, dtype: float64
```

qui invece vediamo i prezzi dei biglietti e sostituiamo i costi mancanti sempre con .fillna calcolando la media dei costi

### 1.1.5 GRAFICO A BARRE SUI SOPRAVVISUTI E I MORTI TRA MASCHI E FEMMINE

[7]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.catplot(x="Sex", hue="Survived", kind="count", data=titanic)
```

[7]: <seaborn.axisgrid.FacetGrid at 0x1aef89e5a90>

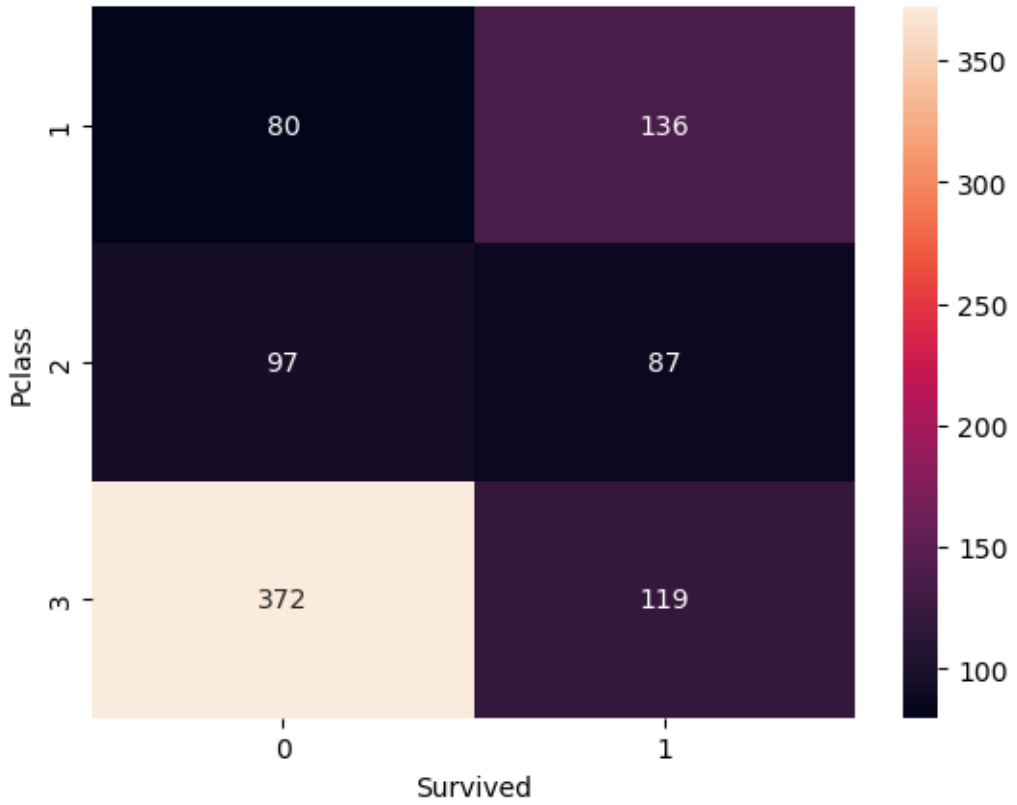



adesso creiamo un grafico a barre prendendo i maschi e le femmine e come valore per l'asse y il numero delle persone e creiamo una colonna blu per il numero di morti e una arancione per il numero dei soppravvisuti

### 1.1.6 CONTROLLO CON HEATMAP DEI MORTI-SOPRAVVISSUTI TRA LE DIVERSE CLASSI

[8]:
```
group = titanic.groupby(['Pclass', 'Survived'])
pclass_survived = group.size().unstack()
sns.heatmap(pclass_survived, annot=True, fmt="d")
```

[8]: <Axes: xlabel='Survived', ylabel='Pclass'>

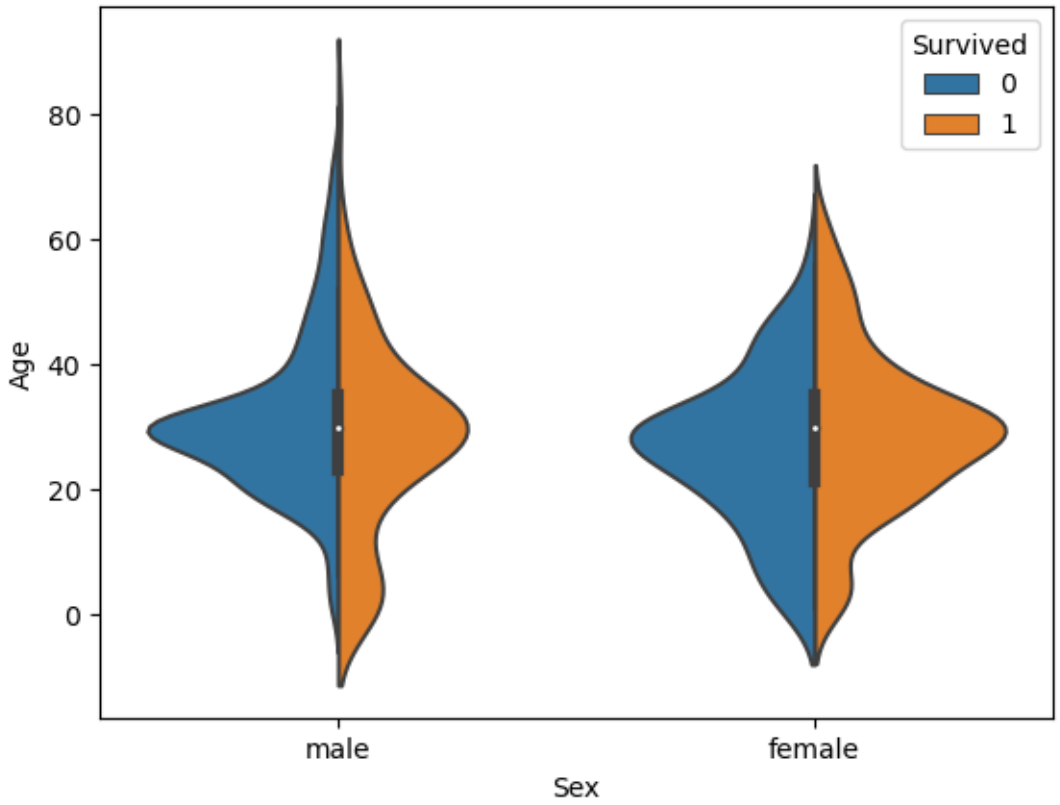



qui abbiamo una heatmap dove è rappresentato sull'asse y la classe di appartenenza e si nota con abbastanza facilità che in terza classe ci sono la maggior parte dei morti dell'intera imbarcazione, in seconda sono abbastamza bilanciati morti e soppravvisuti e invece in prima classe si sono salvate molte più persone rispetto a quelle decedute

### 1.1.7 CONTROLLO CON GRAFICO A VIOLINO DEI MORTI-SOPRAVVISUTI TRA MASCHI E FEMMIEN E LA LORO ETA'

[9]:
```
sns.violinplot(x="Sex", y="Age", hue="Survived", data=titanic, split=True)
```

[9]:
```
<Axes: xlabel='Sex', ylabel='Age'>
```



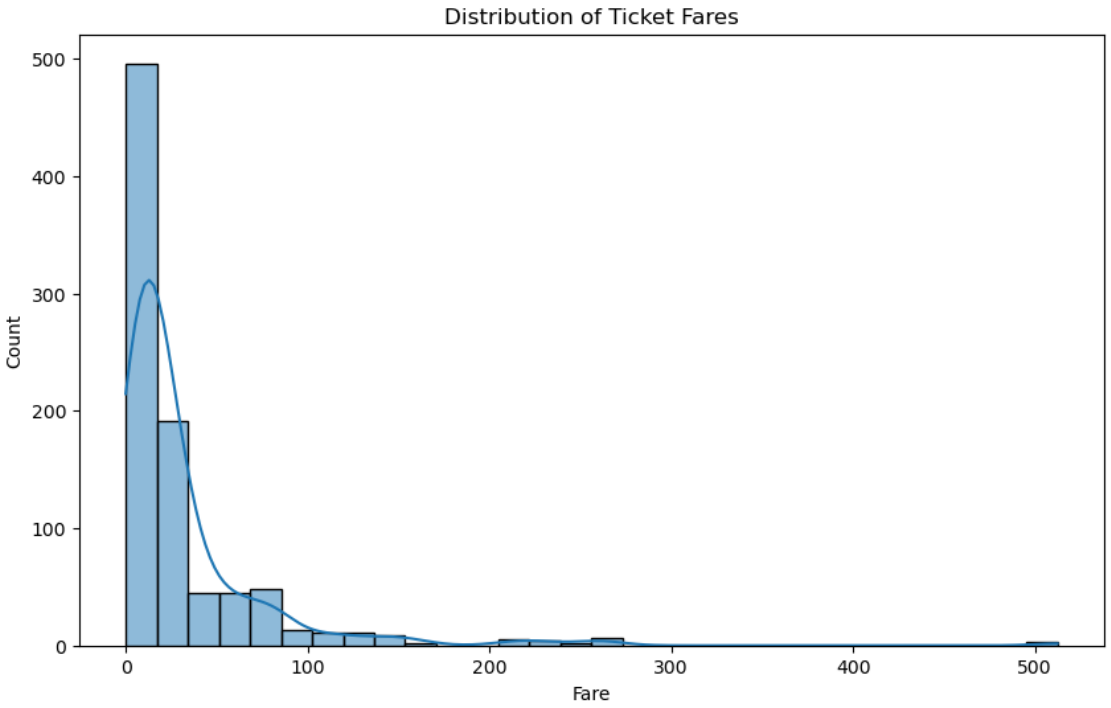

### 1.1.8 CONTROLLO DEI TITOLI DI OGNI PASSEGGERO

[10]:
```
titanic['Title'] = titanic['Name'].str.extract(' ([A-Za-z]+)\.', expand=False)
titanic['Title']
```

[10]:
```
0        Mr
1       Mrs
2      Miss
3       Mrs
4        Mr
       ...
886     Rev
887    Miss
888    Miss
889      Mr
890      Mr
Name: Title, Length: 891, dtype: object
```

### 1.1.9 GRAFICO A DISTRIBUZIONE SUL COSTO DEI BIGLIETTI E DI QUANTI SONO STATI VENDUTI

[11]:
```
plt.figure(figsize=(10, 6))
sns.histplot(titanic['Fare'], bins=30, kde=True)
plt.xlabel('Fare')
plt.ylabel('Count')
plt.title('Distribution of Ticket Fares')
plt.show()
```




qui abbiamo un grafico a distibuzione sui costi e i biglietti venduti

### 1.1.10 CONTROLLO DELLA PERCENTUALE DI SOPRAVVISUTI NELLE DIVERSE FASCE D'ETA'

[12]:
```
titanic['AgeGroup'] = pd.cut(titanic['Age'], bins=[0, 18, 30, 50, 100],␣
 ↪labels=['0-18', '19-30', '31-50', '51+'])
age_survival = titanic.groupby('AgeGroup')['Survived'].mean()
print(age_survival)
```

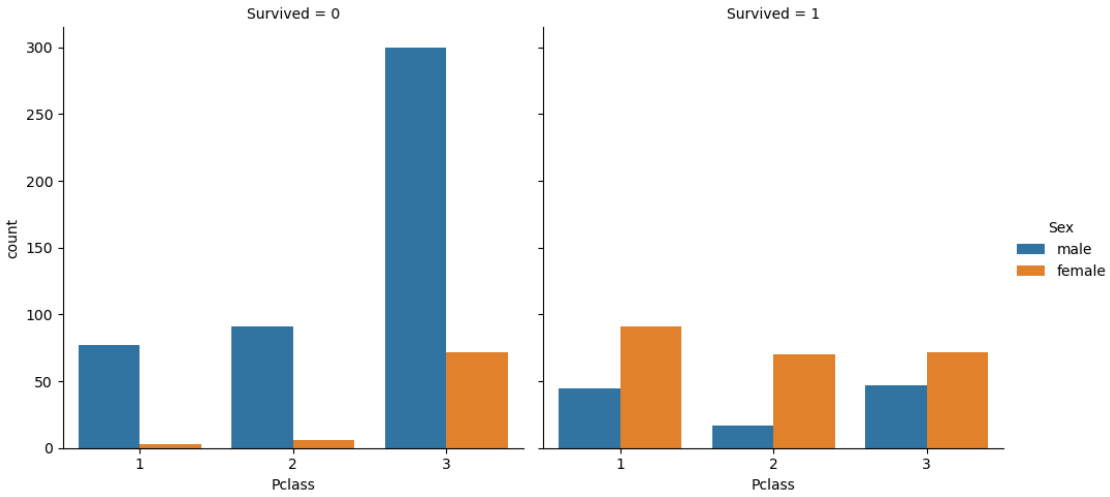
```
AgeGroup
0-18     0.503597
19-30    0.331096
31-50    0.423237
51+      0.343750
```

```
Name: Survived, dtype: float64
```

### 1.1.11 CONTROLLO CON GRAFICO A BARRE DI SOPRAVVISSUTI O MORTI TRA LE DIVERSE CLASSI E IL LORO SESSO

[13]:
```
sns.catplot(x="Pclass", hue="Sex", col="Survived", kind="count", data=titanic)
```

[13]:
```
<seaborn.axisgrid.FacetGrid at 0x1aef92c5fd0>
```




### 1.1.12 PERCENTUALE SOPRAVVISUTI DA OGNI PORTO DI IMBARCO

[14]:
```
embarked_survival = titanic.groupby('Embarked')['Survived'].mean()
print(embarked_survival)
```

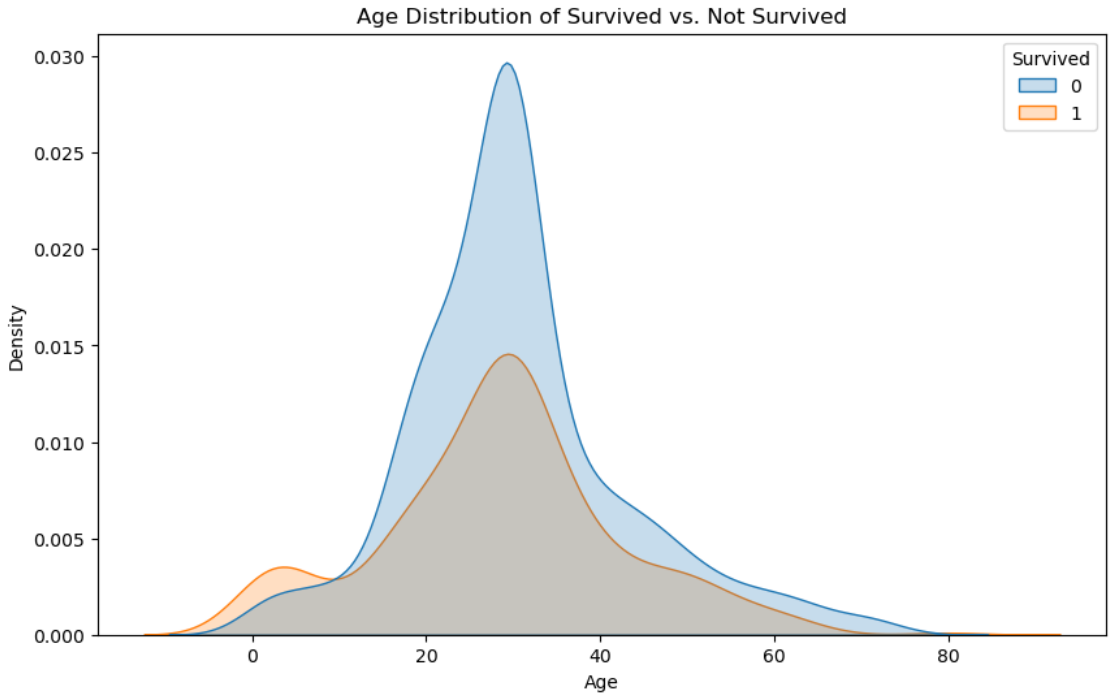
```
Embarked
C    0.553571
Q    0.389610
S    0.336957
Name: Survived, dtype: float64
```

percentuali di sopravvissuti dai diversi porti di imbarco. ( C = Cherbourg ; Q = Queenstown; S = Southampton)

### 1.1.13 GRAFICO DELLA DISTRIBUZIONE DEI SOPRAVVISSUTI O MORTI E LA LORO ETA'

[15]:
```
plt.figure(figsize=(10, 6))
sns.kdeplot(data=titanic, x='Age', hue='Survived', fill=True)
plt.xlabel('Age')
plt.ylabel('Density')
plt.title('Age Distribution of Survived vs. Not Survived')
```

```
plt.show()
```




### 1.1.14 CONTROLLO PERCENTUALE DI SOPRAVVISSUTI PER OGNI TITOLO

[16]:
```
title_survival = titanic.groupby('Title')['Survived'].mean()
print(title_survival)
```

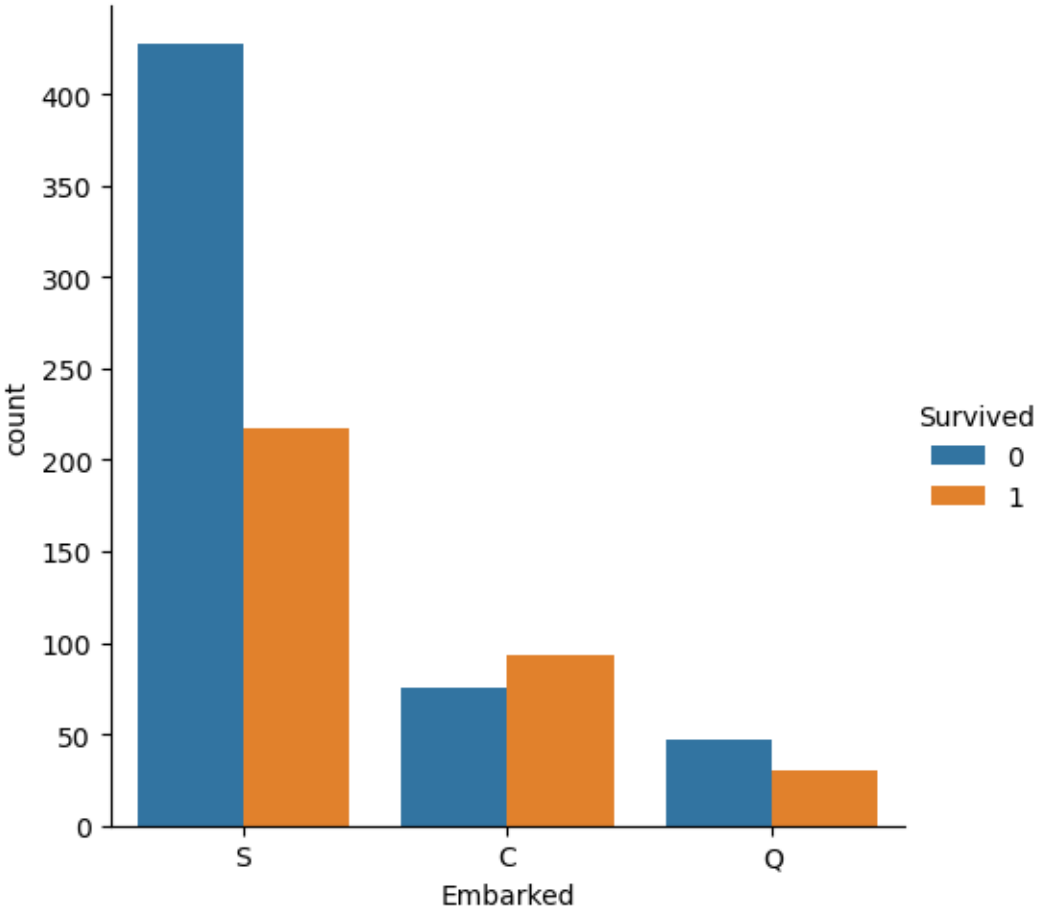
```
Title
Capt        0.000000
Col         0.500000
Countess    1.000000
Don         0.000000
Dr          0.428571
Jonkheer    0.000000
Lady        1.000000
Major       0.500000
Master      0.575000
Miss        0.697802
Mlle        1.000000
Mme         1.000000
Mr          0.156673
Mrs         0.792000
Ms          1.000000
Rev         0.000000
Sir         1.000000
```

```
Name: Survived, dtype: float64
```

### 1.1.15 CONTROLLO MORTI-SOPPRAVVISUTI TRA I PASSEGGERI E DA DOVE SI SONO IMBARCATI

[17]:
```
sns.catplot(x="Embarked", hue="Survived", kind="count", data=titanic)
```

[17]:
```
<seaborn.axisgrid.FacetGrid at 0x1ae80213590>
```




### 1.1.16 CONTROLLO PERCENTUALE SOPRAVVISSUTI TRA LE DIVERSE FASCE DI ETA' E TRA MASCHI E FEMMINE

[18]:
```
age_sex_survival = titanic.groupby(['AgeGroup', 'Sex'])['Survived'].mean()
print(age_sex_survival)
```

```
AgeGroup  Sex
0-18      female    0.676471
          male      0.338028
19-30     female    0.727273
```

```
          male      0.144737
31-50     female    0.779070
          male      0.225806
51+       female    0.941176
          male      0.127660
Name: Survived, dtype: float64
```

### 1.1.17 CONTROLLO PERCENTUALE SOPRAVVIVENZA TRA I DIVERSI TITOLI E DA DOVE SI SONO IMBARCATI

[19]:
```
title_class_survival = titanic.groupby(['Title', 'Pclass'])['Survived'].mean()
print(title_class_survival)
```

```
Title     Pclass
Capt      1         0.000000
Col       1         0.500000
Countess  1         1.000000
Don       1         0.000000
Dr        1         0.600000
          2         0.000000
Jonkheer  1         0.000000
Lady      1         1.000000
Major     1         0.500000
Master    1         1.000000
          2         1.000000
          3         0.392857
Miss      1         0.956522
          2         0.941176
          3         0.500000
Mlle      1         1.000000
Mme       1         1.000000
Mr        1         0.345794
          2         0.087912
          3         0.112853
Mrs       1         0.976190
          2         0.902439
          3         0.500000
Ms        2         1.000000
Rev       2         0.000000
Sir       1         1.000000
Name: Survived, dtype: float64
```

### 1.1.18 CODIFICAZIONE DEL DATAFRAME CON ONEHOTENCODER

[20]:
```
from sklearn.preprocessing import OneHotEncoder

# Seleziona solo le colonne categoriche nel DataFrame
categorical_columns = titanic.select_dtypes(include=['object'])
```

```
# Inizializza OneHotEncoder
encoder = OneHotEncoder()

# Applica l'encoding alle colonne categoriche e trasforma i dati
encoded_categorical_columns = encoder.fit_transform(categorical_columns)

# Ottieni i nomi delle categorie dall'encoder
encoded_categories = encoder.categories_

encoded_column_names = []
for i, col in enumerate(categorical_columns.columns):
    encoded_column_names.extend([f"{col}_{category}" for category in␣
 ↪encoded_categories[i]])

# Crea un nuovo DataFrame con le colonne categoriche codificate e i nomi delle␣
 ↪colonne
df2_encoded = pd.DataFrame(encoded_categorical_columns.toarray(),␣
 ↪columns=encoded_column_names)

# Visualizza il nuovo DataFrame codificato
print(df2_encoded)
```

```
     Name_Abbing, Mr. Anthony  Name_Abbott, Mr. Rossmore Edward  \
0                         0.0                               0.0
1                         0.0                               0.0
2                         0.0                               0.0
3                         0.0                               0.0
4                         0.0                               0.0
..                        …                                 …
886                       0.0                               0.0
887                       0.0                               0.0
888                       0.0                               0.0
889                       0.0                               0.0
890                       0.0                               0.0

     Name_Abbott, Mrs. Stanton (Rosa Hunt)  Name_Abelson, Mr. Samuel  \
0                                      0.0                       0.0
1                                      0.0                       0.0
2                                      0.0                       0.0
3                                      0.0                       0.0
4                                      0.0                       0.0
..                                     …                         …
886                                    0.0                       0.0
887                                    0.0                       0.0
888                                    0.0                       0.0
889                                    0.0                       0.0
```

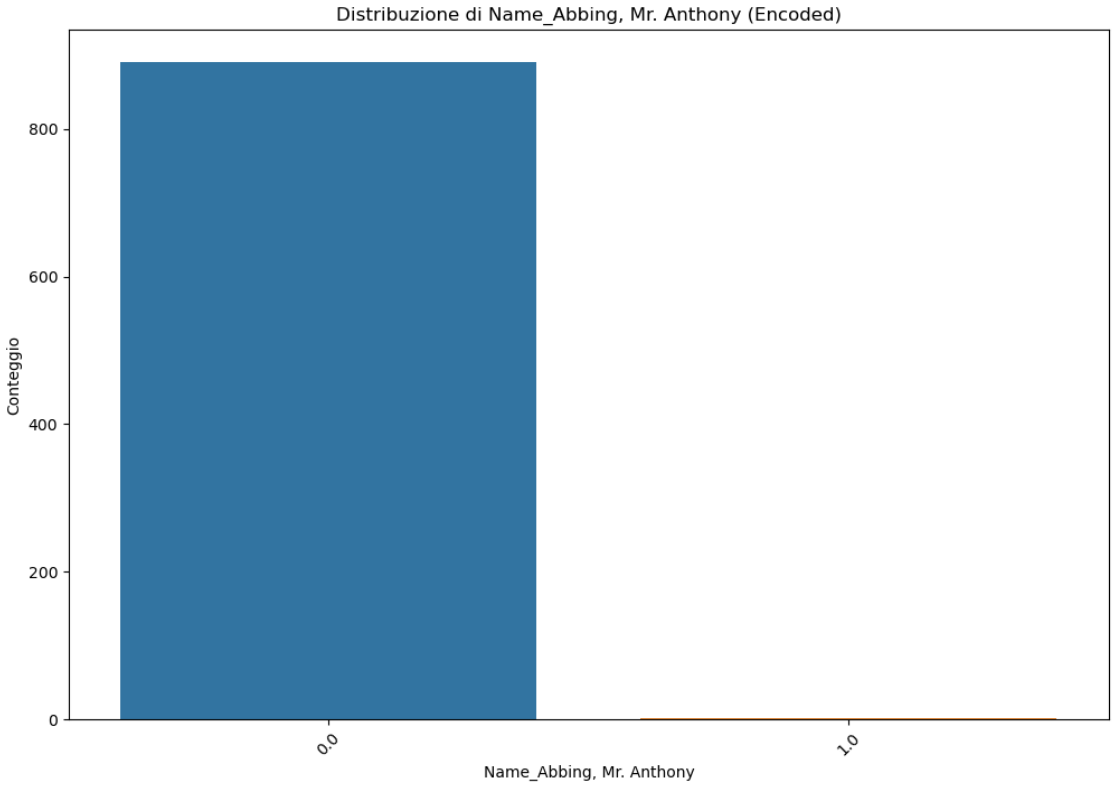
```
890                                   0.0                        0.0

     Name_Abelson, Mrs. Samuel (Hannah Wizosky)  \
0                                           0.0
1                                           0.0
2                                           0.0
3                                           0.0
4                                           0.0
..                                          …
886                                         0.0
887                                         0.0
888                                         0.0
889                                         0.0
890                                         0.0

     Name_Adahl, Mr. Mauritz Nils Martin  Name_Adams, Mr. John  \
0                                    0.0                   0.0
1                                    0.0                   0.0
2                                    0.0                   0.0
3                                    0.0                   0.0
4                                    0.0                   0.0
..                                   …                     …
886                                  0.0                   0.0
887                                  0.0                   0.0
888                                  0.0                   0.0
889                                  0.0                   0.0
890                                  0.0                   0.0

     Name_Ahlin, Mrs. Johan (Johanna Persdotter Larsson)  \
0                                                    0.0
1                                                    0.0
2                                                    0.0
3                                                    0.0
4                                                    0.0
..                                                   …
886                                                  0.0
887                                                  0.0
888                                                  0.0
889                                                  0.0
890                                                  0.0

     Name_Aks, Mrs. Sam (Leah Rosen)  Name_Albimona, Mr. Nassef Cassem  …  \
0                                0.0                               0.0  …
1                                0.0                               0.0  …
2                                0.0                               0.0  …
3                                0.0                               0.0  …
4                                0.0                               0.0  …
..                               …                                 …  …
```

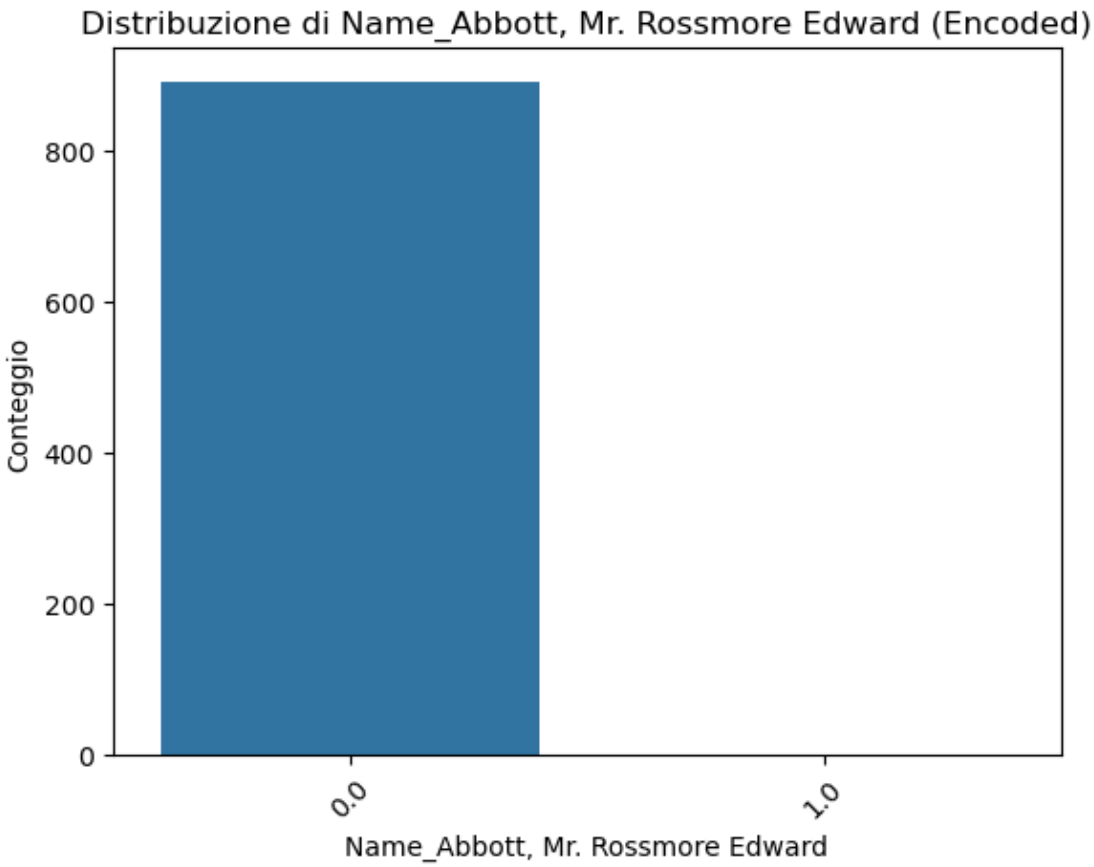
```
886                          0.0                          0.0  …
887                          0.0                          0.0  …
888                          0.0                          0.0  …
889                          0.0                          0.0  …
890                          0.0                          0.0  …

     Title_Major  Title_Master  Title_Miss  Title_Mlle  Title_Mme  Title_Mr  \
0            0.0           0.0         0.0         0.0        0.0       1.0
1            0.0           0.0         0.0         0.0        0.0       0.0
2            0.0           0.0         1.0         0.0        0.0       0.0
3            0.0           0.0         0.0         0.0        0.0       0.0
4            0.0           0.0         0.0         0.0        0.0       1.0
..           …             …           …           …          …         …
886          0.0           0.0         0.0         0.0        0.0       0.0
887          0.0           0.0         1.0         0.0        0.0       0.0
888          0.0           0.0         1.0         0.0        0.0       0.0
889          0.0           0.0         0.0         0.0        0.0       1.0
890          0.0           0.0         0.0         0.0        0.0       1.0

     Title_Mrs  Title_Ms  Title_Rev  Title_Sir
0          0.0       0.0        0.0        0.0
1          1.0       0.0        0.0        0.0
2          0.0       0.0        0.0        0.0
3          1.0       0.0        0.0        0.0
4          0.0       0.0        0.0        0.0
..         …         …          …          …
886        0.0       0.0        1.0        0.0
887        0.0       0.0        0.0        0.0
888        0.0       0.0        0.0        0.0
889        0.0       0.0        0.0        0.0
890        0.0       0.0        0.0        0.0

[891 rows x 1743 columns]
```

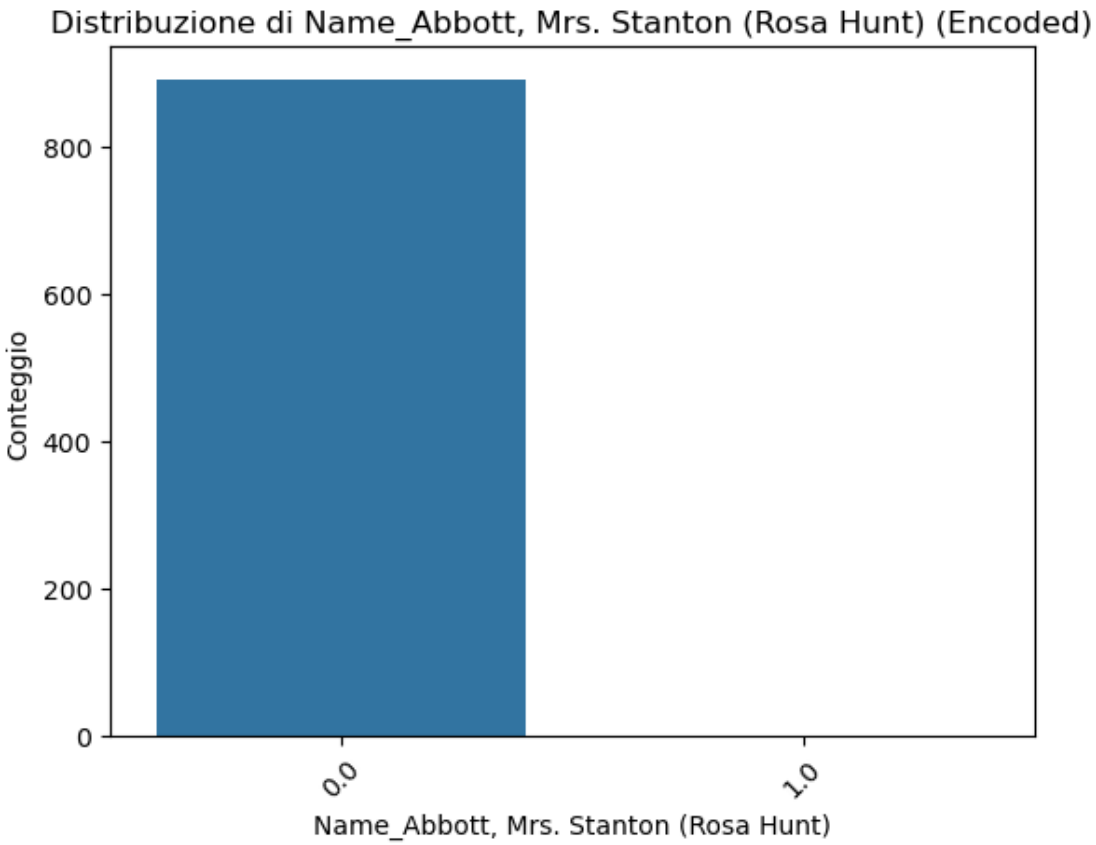
### 1.1.19 ITERAZIONE SUI PRIMI 3 PASSEGGERI SULLA LORO SOPPRAVIVENZA O MORTE

[26]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, OneHotEncoder
# Quindi, abbiamo df2_scaled per le colonne numeriche scalate e df2_encoded per␣
 ↪le colonne categoriche codificate
# Visualizzazione delle distribuzioni delle colonne categoriche codificate␣
 ↪utilizzando grafici a barre
colonne = list(df2_encoded.columns[:3])
plt.figure(figsize=(12, 8))
```

```
for col in colonne:
    sns.countplot(data=df2_encoded, x=col)
    plt.title(f'Distribuzione di {col} (Encoded)')
    plt.xlabel(col)
    plt.ylabel('Conteggio')
    plt.xticks(rotation=45)
    plt.show()
```




Distribuzione di Name_Abbott, Mr. Rossmore Edward (Encoded)
Conteggio
800
600
400
200
0
0.0
1.0
Name_Abbott, Mr. Rossmore Edward




### 1.1.20 CONTROLLO DEL DATAFRAME CODIFICATO PER CONTROLLARE SE E' CODIFICATO CORRETTAMENTE

[22]:
```
df2_encoded
```

[22]:
```
     Name_Abbing, Mr. Anthony  Name_Abbott, Mr. Rossmore Edward  \
0                         0.0                                0.0
1                         0.0                                0.0
2                         0.0                                0.0
3                         0.0                                0.0
4                         0.0                                0.0
..                        …                                  …
886                       0.0                                0.0
887                       0.0                                0.0
888                       0.0                                0.0
889                       0.0                                0.0
890                       0.0                                0.0

     Name_Abbott, Mrs. Stanton (Rosa Hunt)  Name_Abelson, Mr. Samuel  \
0                                      0.0                       0.0
```

```
1                                      0.0                        0.0
2                                      0.0                        0.0
3                                      0.0                        0.0
4                                      0.0                        0.0
..                                     …                          …
886                                    0.0                        0.0
887                                    0.0                        0.0
888                                    0.0                        0.0
889                                    0.0                        0.0
890                                    0.0                        0.0

     Name_Abelson, Mrs. Samuel (Hannah Wizosky)  \
0                                            0.0
1                                            0.0
2                                            0.0
3                                            0.0
4                                            0.0
..                                           …
886                                          0.0
887                                          0.0
888                                          0.0
889                                          0.0
890                                          0.0

     Name_Adahl, Mr. Mauritz Nils Martin  Name_Adams, Mr. John  \
0                                    0.0                   0.0
1                                    0.0                   0.0
2                                    0.0                   0.0
3                                    0.0                   0.0
4                                    0.0                   0.0
..                                   …                     …
886                                  0.0                   0.0
887                                  0.0                   0.0
888                                  0.0                   0.0
889                                  0.0                   0.0
890                                  0.0                   0.0

     Name_Ahlin, Mrs. Johan (Johanna Persdotter Larsson)  \
0                                                    0.0
1                                                    0.0
2                                                    0.0
3                                                    0.0
4                                                    0.0
..                                                   …
886                                                  0.0
887                                                  0.0
888                                                  0.0
```

```
889                                                0.0
890                                                0.0

     Name_Aks, Mrs. Sam (Leah Rosen)  Name_Albimona, Mr. Nassef Cassem  …  \
0                                0.0                                0.0  …
1                                0.0                                0.0  …
2                                0.0                                0.0  …
3                                0.0                                0.0  …
4                                0.0                                0.0  …
..                               …                                  …  …
886                              0.0                                0.0  …
887                              0.0                                0.0  …
888                              0.0                                0.0  …
889                              0.0                                0.0  …
890                              0.0                                0.0  …

     Title_Major  Title_Master  Title_Miss  Title_Mlle  Title_Mme  Title_Mr  \
0            0.0           0.0         0.0         0.0        0.0       1.0
1            0.0           0.0         0.0         0.0        0.0       0.0
2            0.0           0.0         1.0         0.0        0.0       0.0
3            0.0           0.0         0.0         0.0        0.0       0.0
4            0.0           0.0         0.0         0.0        0.0       1.0
..           …             …           …           …          …         …
886          0.0           0.0         0.0         0.0        0.0       0.0
887          0.0           0.0         1.0         0.0        0.0       0.0
888          0.0           0.0         1.0         0.0        0.0       0.0
889          0.0           0.0         0.0         0.0        0.0       1.0
890          0.0           0.0         0.0         0.0        0.0       1.0

     Title_Mrs  Title_Ms  Title_Rev  Title_Sir
0          0.0       0.0        0.0        0.0
1          1.0       0.0        0.0        0.0
2          0.0       0.0        0.0        0.0
3          1.0       0.0        0.0        0.0
4          0.0       0.0        0.0        0.0
..         …         …          …          …
886        0.0       0.0        1.0        0.0
887        0.0       0.0        0.0        0.0
888        0.0       0.0        0.0        0.0
889        0.0       0.0        0.0        0.0
890        0.0       0.0        0.0        0.0

[891 rows x 1743 columns]
```

### 1.1.21 USO DELLA RANDOMFORESTCLASSIFIER

[23]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Definisci la variabile target
target_variable = titanic['Survived']

# Rimuovi la variabile target dal DataFrame codificato
X = df2_encoded
y = target_variable

# Suddivisione dei dati in training e testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Inizializza il classificatore Random Forest
rf_classifier = RandomForestClassifier(random_state=42)

# Addestra il modello sul set di dati di addestramento
rf_classifier.fit(X_train, y_train)

# Effettua le predizioni sul set di dati di test
y_pred = rf_classifier.predict(X_test)

# Valuta le prestazioni del modello
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

# Visualizza l'accuratezza e il report di classificazione
print("Accuratezza del modello:", accuracy)
print("\nReport di classificazione:\n", classification_rep)
```

```
Accuratezza del modello: 0.8156424581005587

Report di classificazione:
               precision    recall  f1-score   support

           0       0.83      0.86      0.85       105
           1       0.79      0.76      0.77        74

    accuracy                           0.82       179
   macro avg       0.81      0.81      0.81       179
weighted avg       0.81      0.82      0.82       179
```

### 1.1.22 USO DELLA SUPPORT VECTOR CLASSIFIER

[24]:
```
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC  # Import Support Vector Classifier
from sklearn.metrics import accuracy_score, classification_report

# Definisci la variabile target
target_variable = titanic['Survived']

# Rimuovi la variabile target dal DataFrame codificato
X = df2_encoded
y = target_variable

# Suddivisione dei dati in training e testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Inizializza il classificatore SVM
svm_classifier = SVC(random_state=42)  # Utilizza SVC invece di␣
 ↪RandomForestClassifier

# Addestra il modello sul set di dati di addestramento
svm_classifier.fit(X_train, y_train)

# Effettua le predizioni sul set di dati di test
y_pred = svm_classifier.predict(X_test)

# Valuta le prestazioni del modello
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

# Visualizza l'accuratezza e il report di classificazione
print("Accuratezza del modello:", accuracy)
print("\nReport di classificazione:\n", classification_rep)
```

```
Accuratezza del modello: 0.8100558659217877

Report di classificazione:
              precision    recall  f1-score   support

           0       0.83      0.85      0.84       105
           1       0.78      0.76      0.77        74

    accuracy                           0.81       179
   macro avg       0.80      0.80      0.80       179
weighted avg       0.81      0.81      0.81       179
```

### 1.1.23 USO DELLA LOGISTICREGRESSION

[25]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

# Definisci la variabile target
target_variable = titanic['Survived']

# Rimuovi la variabile target dal DataFrame codificato
X = df2_encoded
y = target_variable

# Suddivisione dei dati in training e testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Inizializza il classificatore di regressione logistica
log_reg_classifier = LogisticRegression(random_state=42)

# Addestra il modello sul set di dati di addestramento
log_reg_classifier.fit(X_train, y_train)

# Effettua le predizioni sul set di dati di test
y_pred = log_reg_classifier.predict(X_test)

# Valuta le prestazioni del modello
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

# Visualizza l'accuratezza e il report di classificazione
print("Accuratezza del modello:", accuracy)
print("\nReport di classificazione:\n", classification_rep)
```

```
Accuratezza del modello: 0.8212290502793296

Report di classificazione:
               precision    recall  f1-score   support

           0       0.84      0.86      0.85       105
           1       0.79      0.77      0.78        74

    accuracy                           0.82       179
   macro avg       0.82      0.81      0.81       179
weighted avg       0.82      0.82      0.82       179
```